CST8130: Data Structures --- Assign #3- Final Grade Calculator

Using Multiple Generic Data Structures – LinkedList and ArrayList

DUE: Wednesday November 18th by 10PM SHARP!

Problem Description:

In this assignment, you will write a Final Grade Calculator that can be used for any of your courses at Algonquin. We will use multiple Java Generic data structures to hold the data. We could add MUCH to this program – I am providing specifications for the very basics that must be completed for this assignment.

Basic Requirements:

1. Final grades in a course consist of a number of categories (ie Labs, Assignments, Exam, etc.) each of which has a weighting (ie 10%, 25%, 30% , etc) which contribute to a total of 100% . Your program must handle a variable number of categories in a course.
2. Within each category, there will be a number of assessments (the number varies by category). You can assume that each assessment in a category contributes equally to that category's final grade.
3. One execution of this program only needs to handle one course, but multiple students in that course.
4. You must store the grades in a category in a LinkedList (Java generic class). One of the strengths of this data structure is memory efficiency when you don't know how many elements will be in it. (There could be from 1 to 40 or 50 assessment items in a category).
5. You can store the categories in either an ArrayList or a LinkedList (Java generic class). Either is appropriate as we know the maximum number of categories in a course is small – say 10.
6. I have given some suggestions of how you might structure your classes. If you don't use this structure, be sure to ensure you are adhering to all Object Oriented Principles as have been covered in the class.
7. All possible error conditions must be handled. No program abending is tolerated. No System.exit() calls are allowed.

Assessment class:

This class will model a single assessment item with its grade and what that assessment is out of in any category.

Data members:
private String assessmentName;
private float gradeReceived;
private float outOfGrade;

Methods: (this is for you to choose)

GradeCategory Class:

This class models a single category in a grade calculation.

Data members:
private String categoryName;
private LinkedList <Assessment> assessments;
private float categoryWeight;

Methods: (this is for you to choose)

Course Class:

This class will model a course.

Data members:
private String courseName;
private String studentName;
private LinkedList <GradeCategory> allGrades; OR
private ArrayList <GradeCategory> allGrades;

Methods: (this is for you to choose)

Assign3 Class:

This class will contain your method main.

CourseInfo.txt file:

This file contains all the information about a course.

- First line – name of course (note no spaces in the name) and the number of categories for that course
- Remaining lines – one per category consisting of category name, the weight of the category in the final grade, the number of assessments in that category, then for each of the assessments – the name of the assessment and the out of grade

StudentsInfo.txt file:

This file contains all the information about a students in a course.

- First line – name of student (note no spaces in the name)
- Remaining lines – one per category consisting of the grades for each of the assessments for that category

Sample file – CourseInfo.txt:

```
DataStructures 5
HybridActivities 5.0 5 HA1 20  HA2 5
   HA3 5 HA4 5 HA5 5
Labs 10.0 5 Lab1 5 Lab2 5 Lab3 5 Lab4 5
   Lab5 5
Assignments 25 5 Assign1 20 Assign2 20
   Assign3 20 Assign4 20 Assign5 20
Midterms/Tests 30.0  3 Midterm1 30
   Midterm2 10 LabTest 10
FinalExam  30 1 Exam 30
```

Sample file – StudentsInfo.txt:

```
GoodStudent
18 5 5 3 5
5 5 5 5 5
19 17 16 21 15
22 9 8
25
PoorStudent
11 0 0 3 5
5 0 0 5 5
11 5 0 0 0
12 0 0
10
AverageStudent
18 0 4 3 5
5 0 5 5 5
15 15 12 18 12
20 5 8
25
PerfectStudent
20 5 5 5 5
5 5 5 5 5
20 20 20 20 20
30 10 10
30
```

Sample Output

```
Enter name of course file to
process: c:\CourseInfo.txt

Enter name of student file to
process: c:\StudentsInfo.txt

Grades for GoodStudent in
DataStructures
HybridActivities 4.5 / 5.0
Labs 10.0 / 10.0
Assignments 22.0 / 25.0
Midterms/Tests 24.333336 / 30.0
FinalExam 24.999998 / 30.0
Final grade:  85.833336

Grades for PoorStudent in
DataStructures
HybridActivities 2.15 / 5.0
Labs 6.0 / 10.0
Assignments 4.0 / 25.0
Midterms/Tests 4.0 / 30.0
FinalExam 10.000001 / 30.0
Final grade:  26.150002

Grades for AverageStudent in
DataStructures
HybridActivities 3.3 / 5.0
Labs 8.0 / 10.0
Assignments 18.0 / 25.0
Midterms/Tests 19.666668 / 30.0
FinalExam 24.999998 / 30.0
Final grade:  73.96667

Grades for PerfectStudent in
DataStructures
HybridActivities 5.0 / 5.0
Labs 10.0 / 10.0
Assignments 25.0 / 25.0
Midterms/Tests 30.0 / 30.0
FinalExam 30.0 / 30.0
Final grade:  100.0
```

Submission:

You must submit to the assignment link in Blackboard by the due date and time a zip file (named LastnameFirstNameAssign3) containing:

- all source code – ie .java files (Note – I may choose to re-compile your program….so all code must be available to me) with
- Your test plan in either .docx or .xls format

Failure to provide any of the above will have an effect on your grade for this assignment. Marking guide will be published shortly.

Hints:

START THIS ASSIGNMENT TODAY! This assignment needs to be tackled in a structured fashion in order for it to be finished quickly. Do not write more than 20-30 lines of code at a time without running your program. Start with opening and reading from the file to make sure that is working. Then work up in layers from there. Enjoy!!